

Большие почтовые рассылки с помощью йаПосылалка (использование шеллов)

Хотя изначально йаПосылалка не была ориентирована на массовые рассылки, после введения функционала отправки через шеллы ее все чаще стали использовать для таких задач.

По умолчанию йаПосылалка отправляет письма с того сервера, на котором она сама стоит используя его sendmail, но часто у сервера и у почтовых сервисов (вроде mail.ru или gmail) есть ограничения на количества писем. Со среднего сервера можно в сутки отправить 100-200 писем не опасаясь, что он на долгое время попадет в черные списки почтовых сервисов. Это количество существенное, но со временем потребности в почтовом трафике могут расти, и один сервер перестанет их удовлетворять.

Раньше для отправки с множества серверов на каждый требовалось загружать йаПосылалку и вручную вести рассылку. Но это не очень удобно. Современная йаПосылалка (в статье идет речь о версии 2.0.3) может располагаясь на одном сервере рассылать письма через множество серверов. Для этого используются скрипты для выполнения команд на сервере (шеллы) или идущий в комплекте с йаПосылалкой ARS.php (умеет только пересылать письма).

На этом теоретическую часть закончим и перейдем к примерам.

Предположим у нас есть 5 серверов. Это могут быть купленные шеллы (обычно wso2) или свои собственные сервера. Есть 2 пути их использовать:

1. Через уже установленный шелл
2. Через ars.php

Установка пароля

В первую очередь нам надо поставить на шелл или ars наш собственный пароль. И там и там храниться не сам пароль в чистом виде, а только его хэш, который позволяет только проверить правильность введенного пароля, но не узнать его. Открываем wso2 или ars в текстовом редакторе.

Для WSO2:

В самом начале находим строчку

```
$auth_pass = "тут хэш пароля по умолчанию";
```

, нам надо заменить то что находится внутри кавычек. Заходим на любой сервис по генерации MD5 хешей, например <http://mailer.a-l-e-x-u-s.ru/tools/md5/> , вводим желаемый пароль, нажимаем «Посчитать» и заменяем указанный хэш на наш собственный.

Для ARS:

Все похоже на wso2, но ars имеет префикс пароля, что защищает хеш от перебора по базам. Находим строку

\$pass="тут надо будет ввести Ваш хэш";

Заходим на сервис генерации MD5 хешей, например <http://mailer.a-l-e-x-u-s.ru/tools/md5/>, вводим желаемый пароль, но перед ним добавляем "IMAIL" большими буквами без пробела. Нажимаем «Посчитать» и указываем это хэш в ARS.php

Вычисление MD5 хеша

IMAILqazwsx

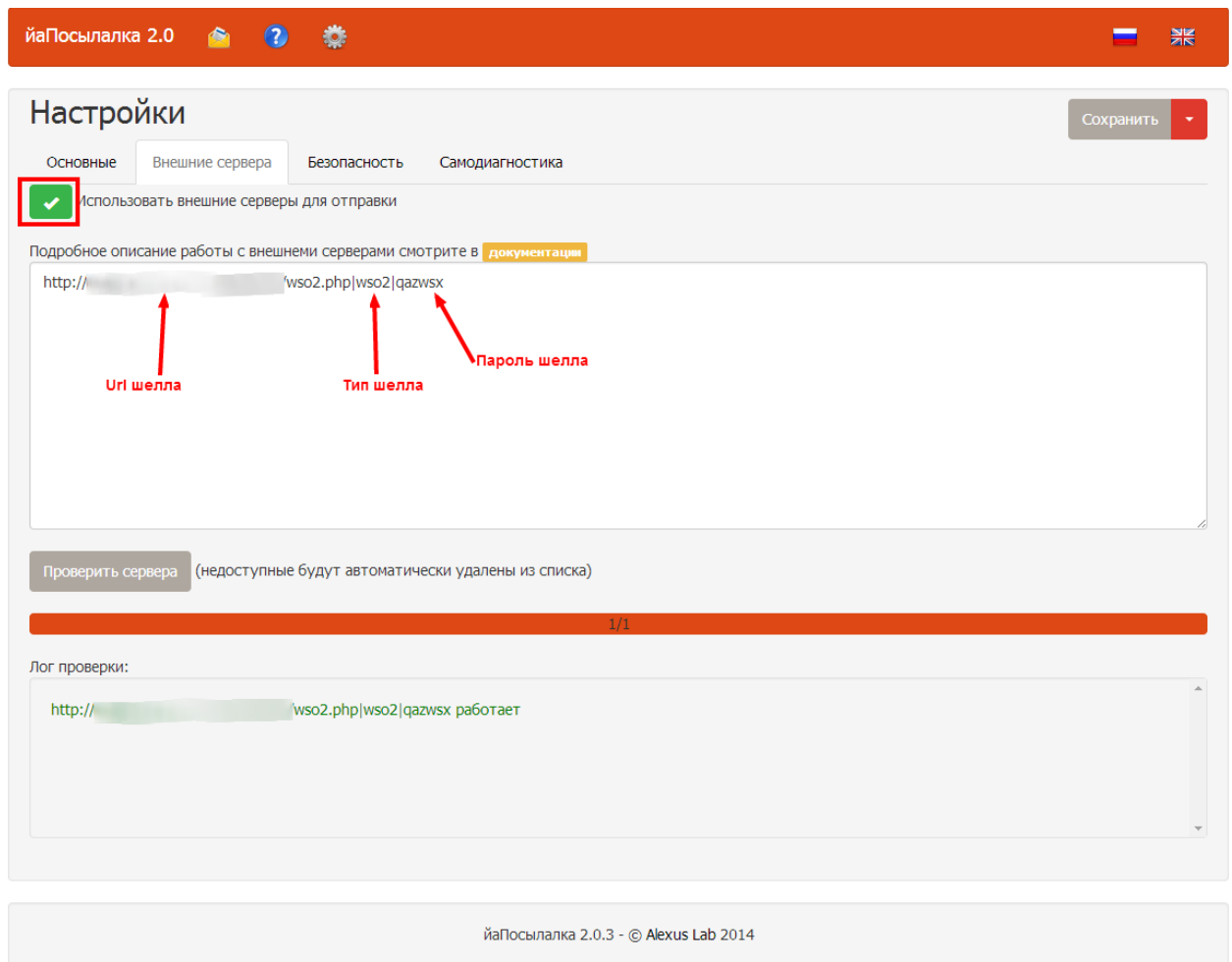
163b388850e190e8903a119aad5783c5

Посчитать

Указание внешних серверов в йаПосылалка

В йаПосылалка открываем настройки – Внешние сервера, и отмечаем что мы будем использовать внешние сервера, далее заполняем наши сервера, по штуке на строчку. Сначала идет адрес, потом тип (wso2 или ars) , потом пароль, разделяется это все символом |.

После заполнения списка серверов, можно провести простую проверку на корректность введенных данных, во время нее проверяется доступность шелла, правильность указанного типа, правильность указанного пароля.



Как используются шеллы и взаимодействуют с потоками

Шеллы всегда используются строго один за одним, в случае одного потока это звучит просто, но если их становится больше, то представить порядок работы уже сложнее.

Предположим у нас есть 5 шеллов и 2 потока.

- 1 Поток берет 1 Шелл
- 2 Поток берет 2 Шелл
- 1 Поток берет 3 Шелл
- 2 Поток берет 4 Шелл
- 1 Поток берет 5 Шелл
- 2 Поток берет 1 Шелл
- 1 Поток берет 2 Шелл

Так же часто используется метод когда количество шеллов равно количеству потоков, при широком канале это может дать максимальную скорость рассылки.

Еще стоит упомянуть как работает задержка. Предположим у нас есть два шелла и нам надо чтобы каждый отправил по 4 письма в час. Для этого устанавливаем 2 потока и задержку 15 минут.

Шеллы не доставляют письма

Не все шеллы одинаково полезны =) . Шеллы необходимо проверять на доставку, отправляя вручную через них письма на целевые почтовые сервисы. Но что делать если у вас уже 100 шеллов

и более? Можно потратить весь день на такую проверку. С версии 2.0.1 в йаПосылалка присутствует макрос [OUTSERVER], который позволяет автоматизировать этот процесс, о нем я расскажу в следующей статье.